REF. AD PCA 88250
EP. 0608 711
COUNTRY US

OPIC
CIPO

(21) (A1) 2,151,023
(22) 1995/06/05
(43) 1995/12/07

(51) Int.Cl. [6] H03M 7/30

(19) (CA) **APPLICATION FOR CANADIAN PATENT** (12)

(54) Method of Coding/Decoding of a Data Stream

(72) von Re[illegible]ntlow, Christian - Germany (Federal Republic of) ;
Grüger, Klaus - Germany (Federal Republic of) ;

(71) SICAN Gesellschaft für Silizium-Anwendungen und CAD/CAT Niedersachsen mbH - Germany (Federal Republic of) ;

(30) (DE) P 44 19 678.4 1994/06/06
(DE) P 44 36 956.5 1994/10/15

(57) 9 Claims

Notice: This application is as filed and may therefore contain an incomplete specification.

ABSTRACT

A method of coding/decoding of a data stream which minimizes the memory requirement for the buffer, the necessary memory band width and the data rate to be stored in the buffer. The method comprises storing the data in the buffer after the data has been compressed and decompressing the data after it is read out.

Method for the coding/decoding of a data stream

Coding and decoding methods are used to minimize the data rate and hence to transmit as much data as possible using a small number of channels or to be able to store as much data as possible in as little memory as possible.

The methods for coding and decoding video and audio information are defined, for example, in the "Joint Picture Expert Group" (JEPG) and "Motion Picture Expert Group" (MPEG) standards. Devices for carrying out these methods are mostly based on digital signal processors (DSP). Single-chip implementations, too, are already commercially available.

Prior art

A data stream consists of a sequence of elements whose data contents can be combined to form various groups. New data contents are allocated to the groups at regular intervals by the data stream. The groups in their entirety form a data record. In the processing of data streams, it is often necessary to buffer data blocks between the individual method steps. In addition, structured data streams often have to be reordered into a different grouping. The memory required for this is dependent on the maximum volume of data which has to be stored in parallel per unit time.

The memory requirement can be reduced by using addressing methods in which, for example, free memory locations are used between data which are stored block by block. Virtual addressing is also possible, in which a logic address which is proportional to the position of the element in the segment is converted into a physical address by means of a transformation function. Nevertheless, the resultant memory requirement is too high for

large volumes of data, such as occur, for example, in digital image processing. Moreover, the problem remains that the data rate at which the information is stored in the physical memory is just as high.

The problems of buffering and reformatting will now be explained in more detail with regard to the example of digital image processing using the MPEG method:

Natural frame sequences have three properties which can be utilized for coding. First of all, successive frames of a frame sequence are similar. Changes from frame to frame can seldom be attributed to new frame contents, instead they can be attributed predominant · · the movement of objects. Furthermore, natural fr... consist, to a first approximation, of areas and edges. In this case, the areas take up the majority of the frame. They are characterized by gradual colour and brightness transitions. Therefore, adjacent pixels within a frame are often similar. Finally, not all grey shades occur with equal probability within natural frames. An average tonal value is more likely to occur than black and white.

These three effects describe the similarity of the signal to itself, namely the similarity of successive frames, the similarity of locally adjacent pixels and the similarity of probability distributions of the tonal values of different frames. These redundancies can be utilized for compression by resolving temporal redundancies of successive frames by means of a movement-compensated DPCM loop (Differential Pulse Code Modulation) and by resolving spacial redundancies and irrelevancies within a frame by means of a DCT (Discrete Cosine Transformation) technique and VLC techniques (Variable Length Coding, consisting of run length coding and Huffmann coding). These techniques for video compression are referred to as hybrid coding. They are already adequately known and proven from video conference and remote contribution insertion codecs, the CCITT H.261 and ITU-T CMTT/2 standards, and from the JPEG method.

The data compression is carried out in an MPEG encoder. In a first stage, the digital video data are initially reordered from the line-by-line frame representation into a sequence of 8 x 8 block matrices (frame reordering), since a complete television picture which can be represented in a two-dimensional matrix is transmitted in such a way that the pixels are combined in a line-by-line manner and the lines succeed one another in time. In this case, the pixels correspond to the elements of a data stream, the elements in their entirety in one line correspond to a group, and the frame corresponds to a data record. The block-oriented coding method described in the MPEG standard envisages that the frames which are normally present after having been organized in a line-by-line manner are reordered, subdivided into blocks and finally processed in groups of blocks, so-called macro blocks. In this case, each of these blocks consists of 8 block lines having 8 pixels each. The groups are converted into a different grouping by splitting a frame which is represented in a line-by-line manner into non-overlapping blocks having a fixed number of columns and lines. In this case, the blocks each represent a frame excerpt. In addition, the temporal succession of the data records - frames - is likewise changed.

A frame store is provided for this reformatting, which store has to be addressed in such a way that it is possible to access the group elements in a specific manner. The groups must therefore have a regular structure. In contrast, the reordering of the data records no longer requires any structure, since access is made to the blocks in their entirety. If the overall available memory size is adequate, the individual block lengths may be variable.

In block-oriented video coding methods, the memory requirement for the line/block reformatting results in n-1 lines for blocks having n lines. In addition to this, there is the memory space for altering the frame order. This memory requirement results from the

number of frames b to be buffered and the frame size. An additional factor is that a plurality of memory banks and address generators are necessary in the case of multistage reformatting. In addition, the information has to be read and written many times.

A second stage is used for data compression by means of the hybrid coding methods mentioned. A frame store is required, in which preceding frames are buffered and used for movement compensation of the respectively current frame. The store is normally arranged downstream of the arithmetic units for carrying out a discrete cosine transformation (DCT) and quantization (Q), as well as inverse quantization (Q^{-1}) and inverse DCT (DCT^{-1}). The video data are transformed and inverse-transformed prior to storage, in order that the coding errors in the movement estimation can be compensated for.

Conventionally, uncompressed frames are stored in the two stores of an MPEG encoder. Consequently, the stores have to be designed to be relatively large. Depending on the standards and frame formats to be processed, the stores for the first and second stages have sizes between about 300 kbytes for 1/4 of a television picture and more than 4 Mbytes for a high-definition television picture (HDTV picture).

The data decompression is carried out in an MPEG decoder. For this purpose, the video data of a frame which have been reordered and coded according to the MPEG standard are read in by an MPEG decoder. In a first stage, the fundamental value details and frame parameters are read from the data stream (header detection) and the data which are coded with a variable code word length are decoded in the variable length decoder (VLD). In this case, the video data are reordered via an inverse zig-zag function into a block arrangement again. A first frame store is used for this purpose. The reordered and VLD-decoded data stream is split up into blocks and further decoded in a second stage by means of dequantization (Q^{-1}) and two-dimensional inverse discrete cosine transformation (DCT^{-1}). The movement compensation is then

reversed. The corresponding stage with a frame store is used for this purpose, which store can also be used as a buffer for the video output.

The first decodable frame of a sequence is a frame (I-picture) which is "intra"-coded completely without any movement compensation. This frame is written directly to the store as the result of the DCT^{-1}. From there, it can be used for movement compensation for the next decoded frame, for example a "predictive" frame (P-picture), which is predicted from preceding frames, or a frame (B-picture) which is "bidirectionally predictive" from preceding and following frames. The decoded P-pictures are likewise buffered for the movement compensation of B-pictures, if the latter occur in the sequence.

In the stage for movement decompensation, the video data, if they are P- or B-pictures, are converted to match the uncompressed video data, located in the store, of a preceding I- or P-picture or of two preceding B-pictures. The I-pictures are directly decompensated without any information from the store. The decompensated data stream corresponds once more to a decoded normal frame having luminance and chrominance values according to the CCIR standard.

The frame store in the second stage is normally addressed with the aid of the VLD-decoded movement vectors. In this case, a displacement position of a macro block of the current frame with respect to the position of a macro block from a preceding frame is indicated in a macro block-by-macro block manner by these vectors.

The decoder also ensures that, with regard to the first frame store for reordering and buffering the data stream, access has to be made to the group elements in a specific manner. In addition, there is a large memory requirement for the frame stores of the first and second stages.

For multichannel audio transmission of audio data, the latter are compressed in an MPEG encoder. The channels of the data stream are subband-filtered in a

first stage. The resultant frequency bands are quantized and matrixed in order to form two compatible stereo channels. The coding of the data streams takes place in a second stage. The data are subsequently formatted in such a way that they can be transmitted sequentially as an MPEG data stream on one channel. For this purpose, the data are split into packets and provided with control information.

Buffers are needed to implement these coding methods. In contrast to the video encoder, the storage problem is not so much the memory size as the memory bandwidth. Specifically, the situation may arise that the data rate of a video data stream is too high. The memory access time is then exceeded. Remedial action is then provided by the cost-intensive use of more rapid memory chips or by minimizing the data rate.

The decompression of audio data streams takes place in an audio decoder. In this case, the data stream is deformatted in a first stage. Corresponding to the video decoding section, the control data are read from the header of a sequence (header detection) and fed together with the coded sample values to the decoding section. The data are converted back into subbands once more at the same time. In a further stage, the channels are recovered by dematrixing and are subband-filtered. The result is channels of pulse-code-modulated audio data (PCM data) which exist in the time domain.

Memory is likewise required for the decompression and, for reasons of space, it is normally used jointly for the various stages. Even here there are not only problems with regard to the memory size but also, primarily, conflicts owing to an excessively small memory bandwidth.

The invention

The invention is based on a method for the coding/decoding of a data stream, which method includes the method stages of reordering as well as compression and decompression of the data, the data being stored in

a buffer for the purpose of reordering and being read from the buffer in a different order. The object of the invention is to improve this method to the extent that the memory requirement for the buffer, the necessary memory bandwidth and the data rate to be stored in the buffer can be minimized, in other words the data streams can be reordered in a manner which saves resources and optimises time. This object is achieved according to the invention by the fact that the data are stored in the buffer after having being compressed and are decompressed again when they are read out.

If it is intended, in the method for the coding/decoding of the data stream, in the stage of compression or decompression, to store data blocks from the data stream in a second buffer and to read them out again, it is envisaged in further pursuance of the concept of the invention that the data are stored in the second buffer after having been compressed and are decompressed again when they are read out.

The volume of data to be buffered is considerably reduced by means of the compression. This results in significant further advantages. In particular, it thus has an advantageous effect in video processing in that considerably smaller memories can now be used. In the case of audio processing, the primary factor is that the number of memory accesses and thus the necessary memory bandwidth are reduced. Considerably faster buffering of the data streams is achieved in any case. In addition, the hardware cost for the memories is considerably less than in the past and only a small additional gate cost is necessary.

~~In the reordering of compressed data records,~~ access to the variable groups of a data record is determined by the compression method. A compression method is preferred in which the structures and sizes of individual data groups as well as the buffering location are defined independently of the data contents, and in which the compression factor of individual data groups is defined, in other words individual groups are compressed

varied are less suitable for buffering in the stage of reordering, since the reformatting of the grouping then has to take place prior to compression. Direct access to individual groups is then no longer possible without having to store additional address information. The consequence of this is that it is not possible to reduce the memory requirement and the access rate for the reformatting to such a large extent using such compression methods.

Use is preferably made of differencial pulse code modulation (DPCM) in the first buffer or in both buffers for compression and decompression. This provides the advantage of lower computational load and a smaller number of necessary gates. In the image processing, it is also possible to employ a two-dimensional predictor, which operates in a block-by-block manner, for the DPCM, with the use of hierarchical addressing, owing to the similarity present in the line structure in blocks and frames. The compression factor and the quality of the transmitted data can be improved thereby.

In a further embodiment, which is particularly suitable for the second buffer, the compression is carried out in such a way that the data are compressed, with the aid of a two-dimensional DCT, a quantization stage (Q) and a VLC, to a fixed size independently of the frame complexity. This has the advantage that the required memory to be made available can be predicted. To read out the data, the decompression takes place using the inverse methods.

The quantization factor can be readjusted in a block-by-block manner by controlling the quantization stage by means of a quantization controller (rate control). This means that the frame store is always utilized completely. It is possible to achieve an optimum between picture quality and memory requirement.

Drawings

Exemplary embodiments of the invention are described with reference to the drawings. In this case,

commercially available chips are referred to by their customary abbreviations and all other circuit components are referred to by numerals. In the drawings:

Figure 1 shows an MPEG encoder with frame stores for reordering and movement compensation

Figure 2 shows a method for memory minimization

Figure 3 shows an MPEG decoder with frame stores for reordering and movement compensation

Figure 4 shows a special Huffmann encoder for carrying out a VLC

Exemplary embodiments

The method can be used in a multiplicity of data processing devices using all conceivable compression algorithms.

The necessary storage capacity is reduced by the fact that the data are compressed prior to storage and decompressed again when they are read out. The resultant storage capacity is calculated as a function of the compression factor k such that: Spk = Spn / k

where Spk: Storage capacity for compressed volume of data

Spn: Storage capacity for uncompressed volume of data

k : Compression factor

The compression factor k is dependent on the selected compression method and the desired quality of the data to be stored.

The invention will now be explained in more detail with regard to the example of video data processing using the MPEG method. Figure 1 illustrates the structure of a conventional MPEG decoder. The uncompressed data 1 are reordered in an arithmetic unit 2 (frame reordering), a frame store 3 being used for writing 4 and reading 5. The store is controlled by a control unit by means of control signals 6. The following movement estimator 7 accesses the second frame store 8. The uncompressed data of preceding frames are stored there. Compression by means of DCT and quantization Q

then follows. On the one hand, the data are then subjected to variable length coding (VLC), are buffered in a buffer 9 for the purposes of their output and are output as a compressed data stream 10. On the other hand, inverse quantization and inverse DCT are carried out and the data are stored in the frame store 8. The quantization factor can be readjusted in a block-by-block manner by means of a quantization controller 11 (rate control).

The method according to the invention can be inserted into the first stage for reordering and the second stage for movement compensation. The areas of use of the invention in the encoder are illustrated by a broken block 12.

The reordering can be carried out in one step if a fixed compression factor for individual groups is specified independently of the data contents. This can be achieved by the use of suitable source coding methods. A possible method is DPCM with a fixed code word length, which is subject to losses and is illustrated in Figure 2, or appropriate transformation coding.

Figure 2 illustrates a block 12 for memory minimization. Instead of the uncompressed storage of the original video data in the memory, the said data are subjected to DPCM. For this purpose, the data stream 4 is converted with data which are predicted by a predictor 13 and is quantized (Q). The data stream is subsequently coded using a Huffmann method 14 and written to the memory 15. For reading out, the data stream is decoded in a Huffmann decoder 16 and converted with the values predicted by means of the predictor 17.

This method of coding and decoding by means of DPCM has the advantage that the compression factor can be defined and, consequently, access can be made to the data in the memory in a specific manner. In addition, the gate complexity for implementation in a semiconductor chip is quite low.

For separate block line coding/decoding, from experience a compression factor of two can be used for

reordering, without substantial impairment to the picture quality occurring. It is possible directly to access any block line by means of hierarchically structured addressing. This means that all reformatting operations can be carried out in one step, after writing to the memory 15, which should be implemented as RAM with random access, by means of direct addressing of the block lines when they are read out.

Since each group has been compressed without regard to adjacent groups, the signal can subsequently be decompressed again separately.

Another exemplary embodiment of the invention is illustrated in Figure 3. The data stream 4 is subjected to two-dimensional discrete cosine transformation (DCT) and is subsequently quantized by means of a quantization stage (Q). The quantized date 18 are compressed in a special Huffmann coding section (VLC), which is illustrated in Figure 4. The compressed data stream 19 is then stored in the memory 15. A quantization controller 20 (rate control) undertakes the control of the quantization. When the compressed data are read out, these steps are then reversed by means of the corresponding inverse operations (VLD, Q^{-1}, DCT^{-1}). In this case, this method, which is similar to the JPEG standard, has the following special features:

a) Frames are only intra-coded.
b) Frames are compressed to a fixed size independently of the frame complexity.
c) The quantization controller 20 operates predictively, that is to say that during the frame compression it also takes account of the volumes of data which are still to be expected from the as yet unprocessed remaining bits.
d) The Huffmann tables are specially adapted as a function of the compression rate, which is determined in turn by the application.
e) If an uncoded block (8 x 8 pixels) contains less data than a coded block, it is stored uncoded.

Any desired architecture can be used for the DCT,

provided that the accuracy requirements in accordance with the underlying standard (for example MPEG or H.261) are met.

The quantization operates essentially in an analogous manner to JPEG and also has the same two-stage structure. The quantization controller 20 readjusts the quantization factor in a block-by-block manner. This is done as a function of the occupancy of the frame store and of the frame complexity to be expected.

A special Huffmann encoder is illustrated in Figure 4. The quantized data are initially read from a first RAM memory 21 using a zig-zag scan. The first value, the DC component (DC) in the frequency range, is written uncoded to the output memory 22. The subsequent AC components (AC values) are coded in an analogous manner to the MPEG standard using the run length method. The value pairs produced thereby are then transformed by means of a Huffmann table into a bit code of variable length. A Huffmann encoder 23, a multiplexer 24 and a register memory 25 are used for this purpose. It can be discerned at the overflow of the output memory 26 whether the coded block contains more data than the uncoded block. The block is than output after having been DCT-transformed but not, however, compressed.

Therefore, there appears at the output of the VLC a bit stream having coded and also, partly, uncoded data, which are parallelized by a multiplexer 27 in accordance with the word width of the memory and are stored in the register memory 22. Furthermore, a flag must be set for each block in a further memory, which flag indicates whether the block is uncoded or coded. In addition, the quantization factor associated with the block must be stored.

It can be ensured by means of algorithms which are dependent on the application that the frame store is always utilized completely, with the result that an optimum is established between picture quality and memory requirement. The quantization controller 20 together with the quantization section (Q) and the Huffmann coding

section (VLC) determine the compression factor.

The decoding of the frame store data using the inverse method takes place in an analogous manner to the VLC. In this case, the flag which serves to indicate whether a block is coded or uncoded has to be evaluated.

The dequantization operates in an analogous manner to the quantization. The 12-bit-wide input data for the inverse DCT and produced.

Any hardware can be used for the inverse discrete cosine transformation (DCT^{-1}), provided that the requirements of the respective standard are upheld.

The use of the invention is additionally illustrated in Figure 5 with regard to an MPEG video decoder.

In this case, the method can be used, on the one hand, for the reordering of the data stream after the header detection 28. The buffer 29 which is normally used for this purpose can be replaced by the method 12 according to the invention.

The method for movement compensation can be used advantageously. The video data which have been decoded by means of VLD, Q^{-1} and DCT^{-1} are compressed, if they are I- or P-pictures, according to the method shown in Figure 3 and are written to the frame store 30. They are available there for a future P- or B-picture to be decoded. In this way, the last two P-pictures or one I- and one P-picture are always available in the frame store after having been compressed. The data are decompressed again by means of the inverse method and are fed to the decompensation stage 31. The B-pictures are not buffered but rather forwarded from the decompensation stage 31 directly via a multiplexer 32 to the output unit.

The store 30 is split into two partial areas which each have two banks. The memory addresses are generated by the vectors of the VLD and stored in an address memory 33.

The following unit must be able to receive the video data in blocks and not in lines, as is necessary in the case of monitors.

the case of monitors.

The estimated hardware expenditure with and without memory minimization is given below by way of comparison for an MPEG decoder, for two CCIR 601 standard frames, with a 4:2:0 chroma format:

If a compression factory of k-3 is selected for the frame store data for the highest picture quality, then only 44/3 Mbytes are transmitted averaged over 1/50 s. This corresponds to a bit rate of 14.7 Mbyte/s. The memory size of the frame store required for this is then: 720 frame columns x 288 frame lines x (8 + 4 bits per pixel) x 4 memory banks 2 I- and 2 P-pictures) / k - approximately 10/k Mbits = approximately 3.3 Mbits.

The start addresses together with the quantization factor and the flag for coded or uncoded data have to be stored for each block in an address memory. Under the conditions stipulated above, the resulting word width of the address for the frame store is 26 bits per block.

The number of required entries in the address memory depends on the number of blocks, the block size and the standard of the frame: 720 frame columns / 8 (horizontal block size) x 288 frame lines / 8 (vertical block size) x 1.5 (luminance and chrominance) = 4860 blocks per memory bank. There follows from this a total number of 19,440 entries, each of 26 bits, that is to say approximately 505 kbits for the address memory.

An additional expenditure on hardware is necessary for the special driving of the frame store in an MPEG decoder, which hardware comprises a two-dimensional DCT, a quantizer (Q) and a Huffmann encoder (VLC) including an quantization controller 20. For reading from the frame store, an inverse DCT (DCT^{-1}), a dequantizer (Q^{-1}) and a Huffmann decoder (VLD) are required for each partial store.

Hardware to be implemented	Memory minimization	
	with	without
2 DCT^{-1} (60 MHz)	24,000 gates	0 gates
2 Q^{-1}	10,000 gates	0 gates
2 VLD	10,000 gates	0 gates
1 VLC	8,000 gates	0 gates
1 Q	5,000 gates	0 gates
1 DCT (16 MHz)	8,000 gates	0 gates
1 rate control	5,000 gates	0 gates
Address memory	505 kbits	0 bits
Frame store	3.3 Mbits	10 Mbits
TOTAL	70,000 gates + approximately 3.8 Mbits	10 Mbits

Overall, therefore, the resulting memory requirement for the frame store is 3.3 Mbits (data memory) + 0.5 Mbit (address memory) = approximately 3.8 Mbits, compared with 10 Mbits for the uncompressed storage of frames in CCIR 601, 4:2:0.

The hardware expenditure could possibly be reduced further in the event of time-division multiplex operation of individual modules (for example DCTs, Qs) in conjunction with the use of high-speed technologies. However, this is dependent on the individual application and on the limiting frequency of the technology compared with the standard-dependent volume of data to be processed.

When the described method for movement compensation is used, accumulation of the errors produced by the said compression is normally retained and carried over to the next I-picture. If this is not desired, it is possible to limit the error propagation to the successive B-pictures using the following method:

First of all, an I- or P-picture is written uncompressed to the store. The subsequent incoming I- or P-picture is likewise stored uncompressed for the first

64 lines, depending on the maximum length of the movement vector, and is decompensated with the uncompressed I-/P-picture. Then, at the same time, the procedures of reading out the I-picture, compressing it and rewriting it to the store at another location are begun.

The compression in the method according to the invention can also be carried out without any transformation. The video data can be compressed in the space domain by means of the Shannon Entropy Theorem without being converted into the frequency domain by means of DCT/DCT^{-1} or other transformations. For this purpose, an optimum Huffmann table of the data is initially calculated in the space domain, using the probability distribution of the brightness and chrominance values in the space domain. In this case, the Huffmann table is either determined online for each frame excerpt or is established definitely by prior calculation. The advantage of this method is less compression complexity. In order to guarantee a fixed output data rate, this method must be extended by a suitable quantization stage.

WHAT IS CLAIMED IS:

1. Method for the coding/decoding of a data stream, including the method stages of reordering as well as compression and decompression of the data, the data being stored in a buffer for the purpose of reordering and being read from the buffer in a different order, characterized in that the data are stored in the buffer after having been compressed and are decompressed again when they are read out.

2. Method according to Claim 1, data blocks from the data stream being stored in a second buffer and read out again in the stage of compression and decompression, characterized in that the data are stored in the second buffer after having been compressed and are decompressed again when they are read out.

3. Method according to Claim 1, characterized by a compression method in which the structures and sizes of individual data groups as well as the buffering location are defined independently of the data contents and the compression factor of individual data groups is defined.

4. Method according to Claim 1, 2 or 3, characterized in that differential pulse code modulation (DPCM) is used for the compression and decompression.

5. Method according to Claim 4, characterized in that the differential pulse code modulation (DPCM) is carried out using two-dimensional predictors (13 and 17) which operate in a block-by-block manner.

6. Method according to one of Claims 4 and 5, characterized in that the memory locations are assigned by means of hierarchically structured addressing.

7. Method according to Claim 2, characterized in that the data are subjected to a two-dimensional discrete cosine transformation (DCT), are then stored in the memory after having been compressed by means of a quantization stage (Q) and a Huffmann coding section

(VLC) and are decompressed again when they are read out by means of the corresponding inverse operations (VLD, Q^{-1}, DCT).

8. Method according to Claim 7, characterized in that the quantization stage (Q) is controlled by a predictively operating quantization controller (20).

9. Method according to Claim 1, 2 or 4, characterized in that one-dimensional transformation coding is used for the compression and decompression.

2151023

1/5

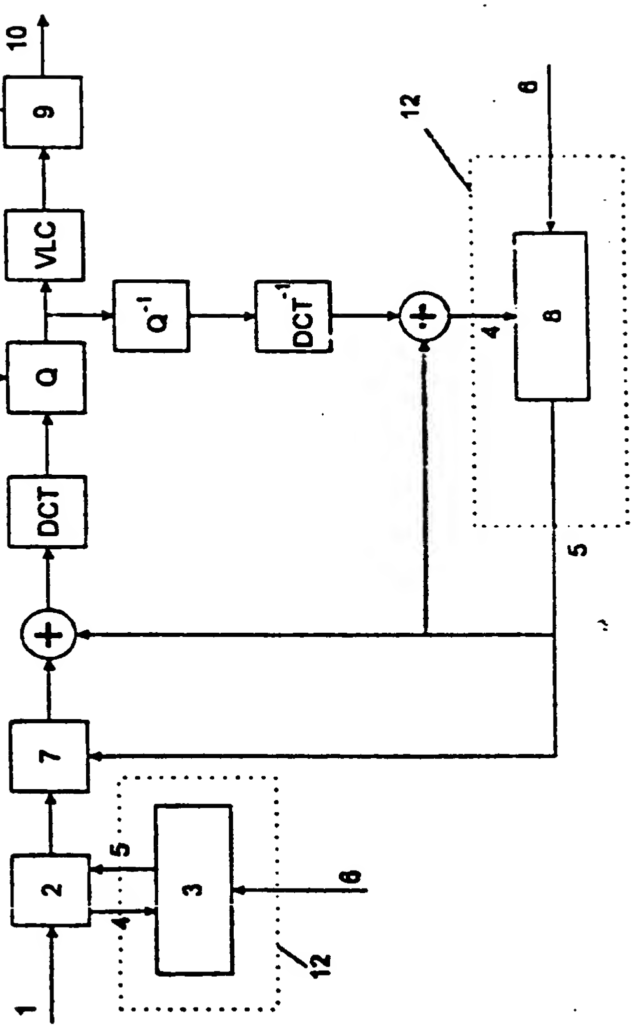


Fig. 1

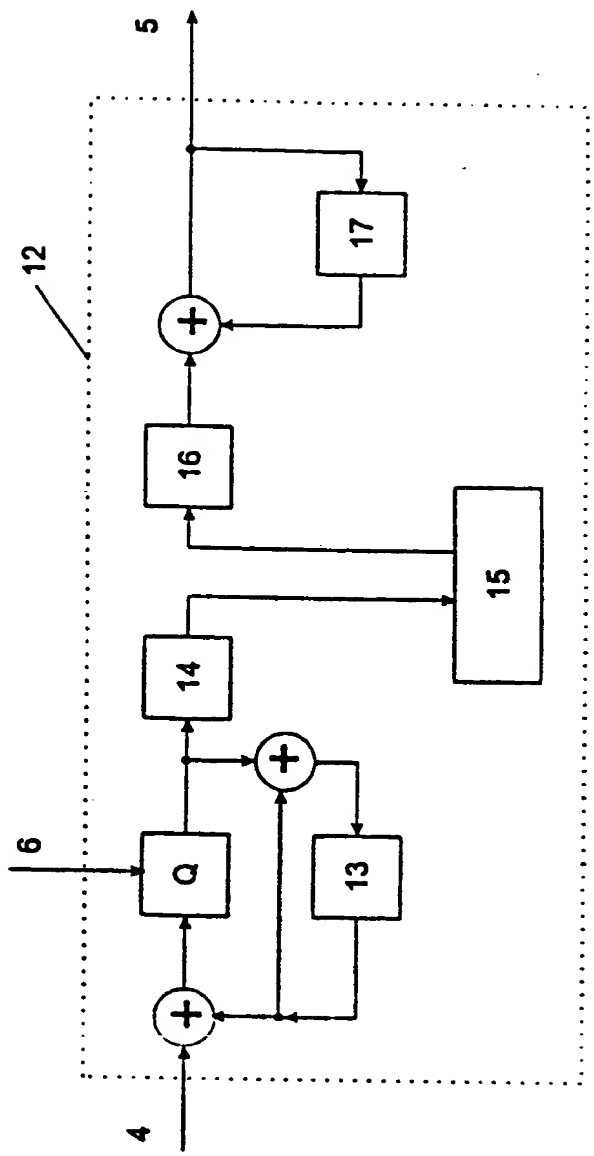


Fig. 2

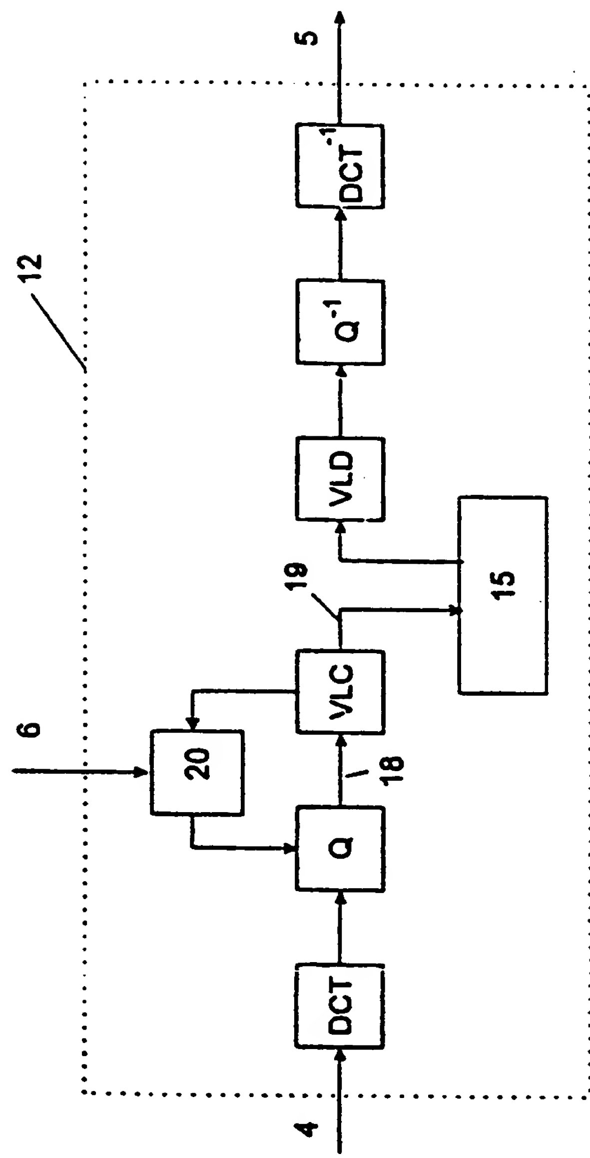


Fig. 3

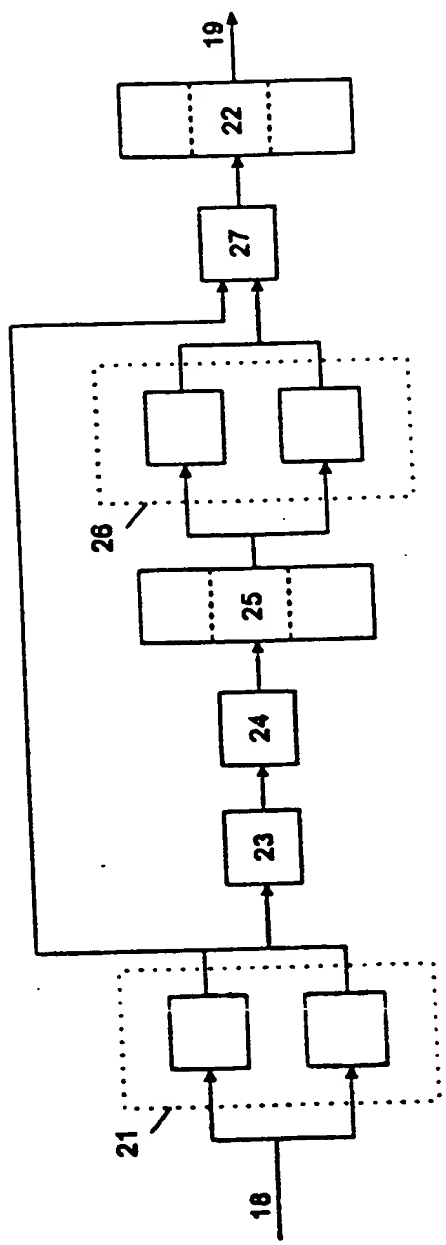
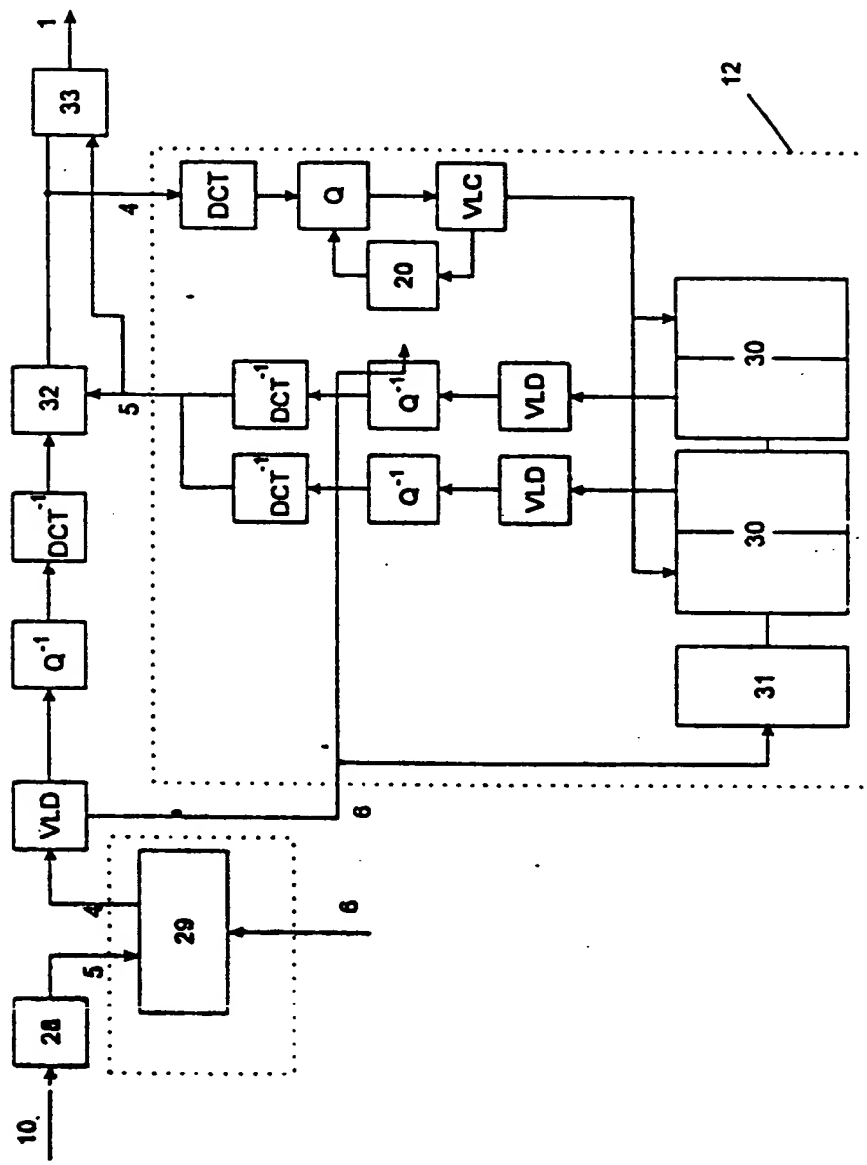


Fig. 4



1
33
32
DCT
Q
VLC
20
DCT⁻¹
Q⁻¹
VLD
DCT⁻¹
Q⁻¹
VLD
30
30
31
12
4
5
6
Q⁻¹
DCT⁻¹
VLD
29
28
10

Fig. 5